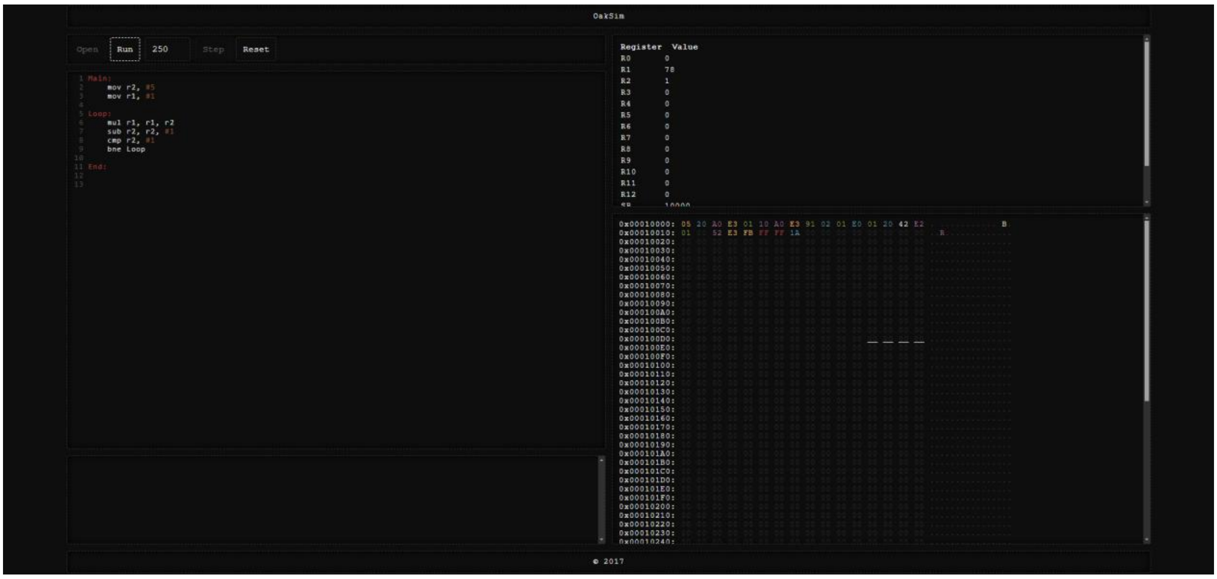
Template Week 4 – Software

Student number: 568403 – Kiarash Delavar – SR

Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:



```
Main:
    mov r2, #5
    mov r1, #1

Loop:
    mul r1, r1, r2
    sub r2, r2, #1
    cmp r2, #1
    bne Loop

End:
```

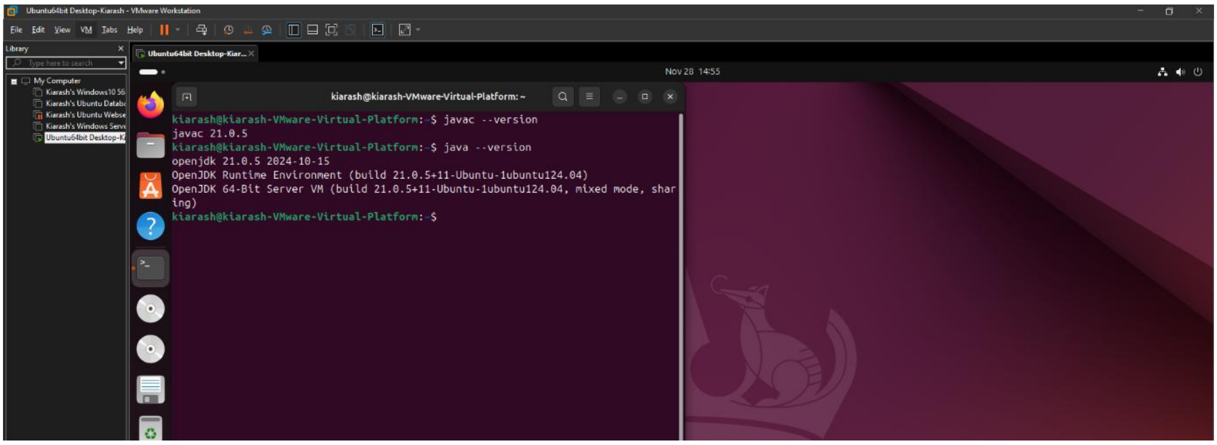
```
Register  Value
R0        0
R1        78
R2        1
R3        0
R4        0
R5        0
R6        0
R7        0
R8        0
R9        0
R10       0
R11       0
R12       0
SP        10000
```

```
0x00010000: 05 20 A0 E3 01 10 A0 E3 91 02 01 E0 01 20 42 E2  ...............B.
0x00010010: 01 00 52 E3 FB FF FF 1A
```

Assignment 4.2: Programming languages

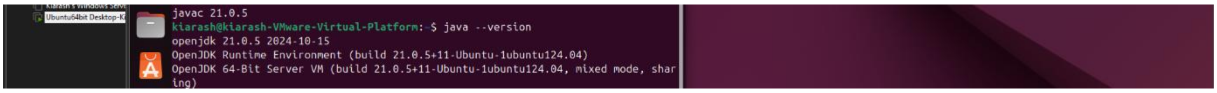
Take screenshots that the following commands work:

javac –version



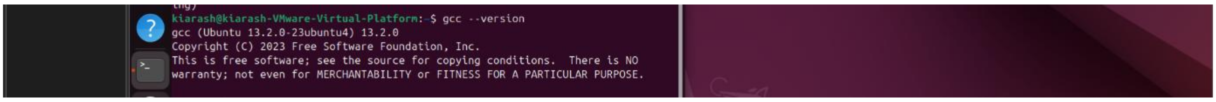
```
kiarash@kiarash-VMware-Virtual-Platform:~$ javac --version
javac 21.0.5
kiarash@kiarash-VMware-Virtual-Platform:~$ java --version
openjdk 21.0.5 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, shar
ing)
kiarash@kiarash-VMware-Virtual-Platform:~$
```

java –version:



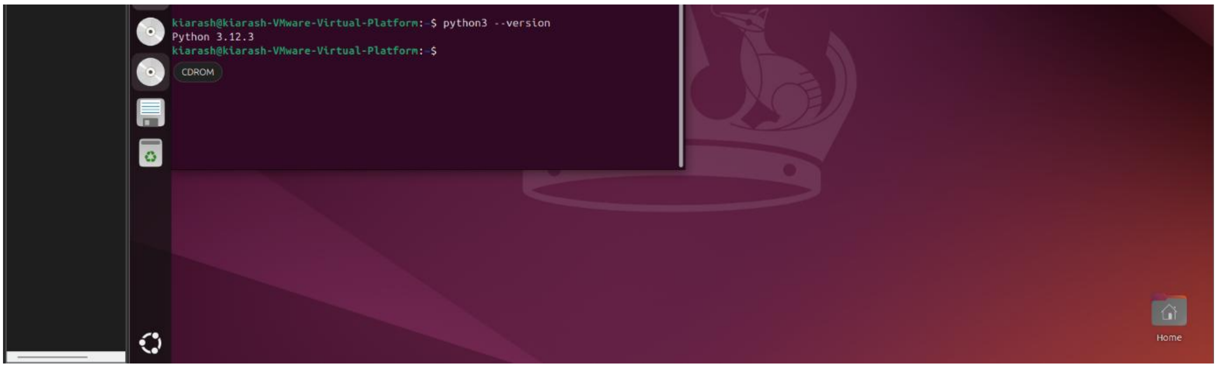
```
javac 21.0.5
kiarash@kiarash-VMware-Virtual-Platform:~$ java --version
openjdk 21.0.5 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, shar
ing)
```

gcc –version:



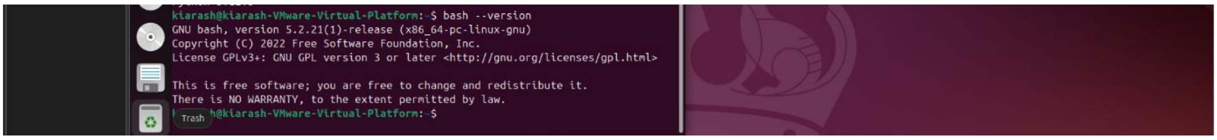
```
ing)
kiarash@kiarash-VMware-Virtual-Platform:~$ gcc --version
gcc (Ubuntu 13.2.0-23ubuntu4) 13.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

python3 –version:



```
kiarash@kiarash-VMware-Virtual-Platform:~$ python3 --version
Python 3.12.3
kiarash@kiarash-VMware-Virtual-Platform:~$
```

bash –version:



```
kiarash@kiarash-VMware-Virtual-Platform:~$ bash --version
GNU bash, version 5.2.21(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
h@kiarash-VMware-Virtual-Platform:~$
```

Assignment 4.3: Compile

Which of the above files need to be compiled before you can run them?

+Fibonacci.java and **fib.c** need to be compiled before running.

Which source code files are compiled into machine code and then directly executable by a processor?

+fib.c is compiled into machine code via a C compiler such as gcc and results in an executable file.

Which source code files are compiled to byte code?

+Fibonacci.java is compiled into byte code using **javac**, producing a **.class file** that can be executed by the Java Virtual Machine.

Which source code files are interpreted by an interpreter?

+fib.py (Python source code) and **fib.sh** (Bash script) are interpreted directly by their respective interpreters (python3 for Python and bash for shell scripts).

These source code files will perform the same calculation after compilation/interpretation. Which one is expected to do the calculation the fastest?

+fib.c is expected to perform the fastest because it is compiled into machine code, which executes directly on the processor without intermediate interpretation.

How do I run a Java program?

+Steps:

1. Compile the file: **javac Fibonacci.java**
2. Run the compiled program: **java Fibonacci**

How do I run a Python program?

+Command: python3 fib.py

How do I run a C program?

+Steps:

1. Compile the file: **gcc fib.c -o fib**
2. Run the compiled program: **./fib**

How do I run a Bash script?

+Command: bash fib.sh or ./fib.sh (For permissions: using chmod +x fib.sh).

If I compile the above source code, will a new file be created? If so, which file?

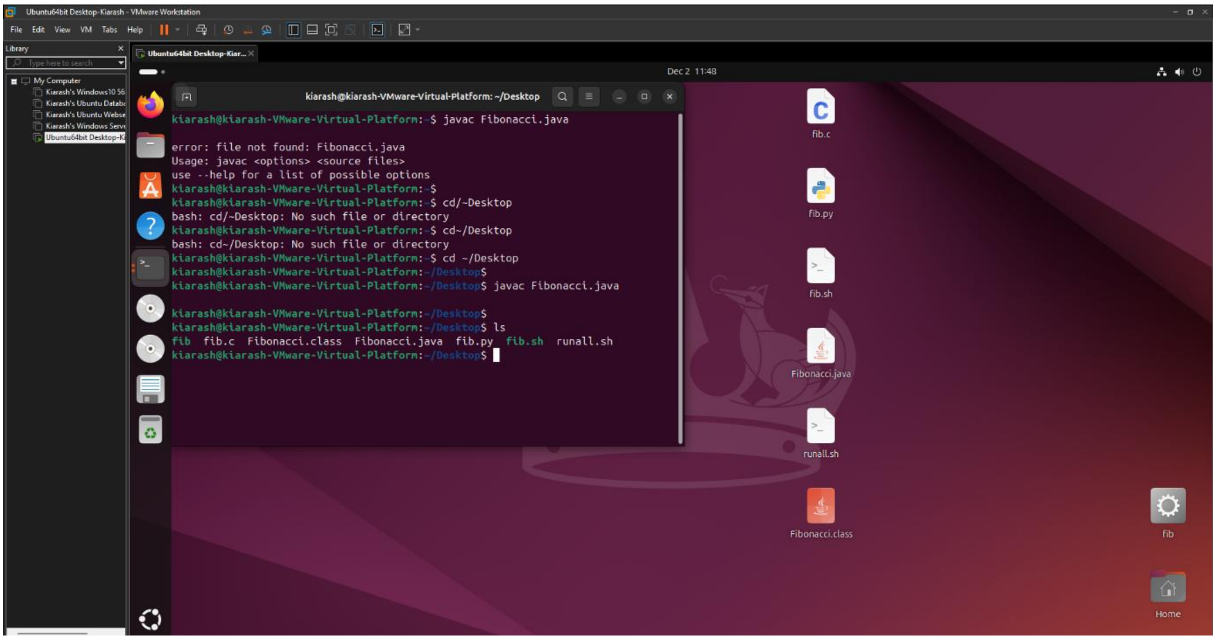
Yes:

- **Compiling Fibonacci.java creates Fibonacci.class.**
- **Compiling fib.c creates an executable file (like , fib if the -o option is used).**

Take relevant screenshots of the following commands:

- **Compile the source files where necessary:**

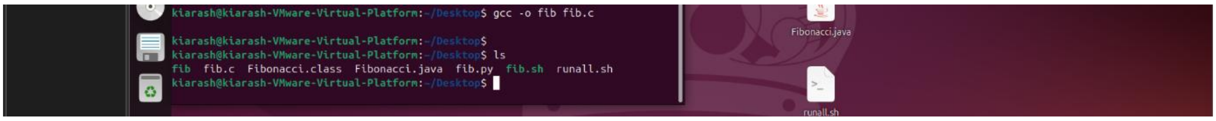
For Fibonacci.java:



Ubuntu64bit Desktop-Kiarash - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
Ubuntu64bit Desktop-Kiar...
Dec 2 11:48
kiarash@kiarash-VMware-Virtual-Platform: ~/Desktop
fib.c
fib.py
fib.sh
Fibonacci.java
runall.sh
Fibonacci.class
fib
Home

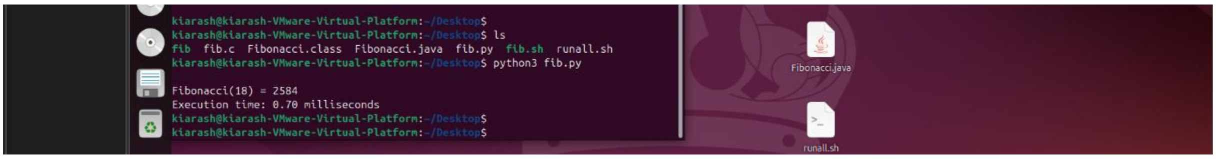
```
kiarash@kiarash-VMware-Virtual-Platform:~$ javac Fibonacci.java

error: file not found: Fibonacci.java
Usage: javac <options> <source files>
use --help for a list of possible options
kiarash@kiarash-VMware-Virtual-Platform:~$
kiarash@kiarash-VMware-Virtual-Platform:~$ cd/~Desktop
bash: cd/~Desktop: No such file or directory
kiarash@kiarash-VMware-Virtual-Platform:~$ cd~/Desktop
bash: cd~/Desktop: No such file or directory
kiarash@kiarash-VMware-Virtual-Platform:~$ cd ~/Desktop
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ javac Fibonacci.java

kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
fib  fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ 
```

For fib.c:



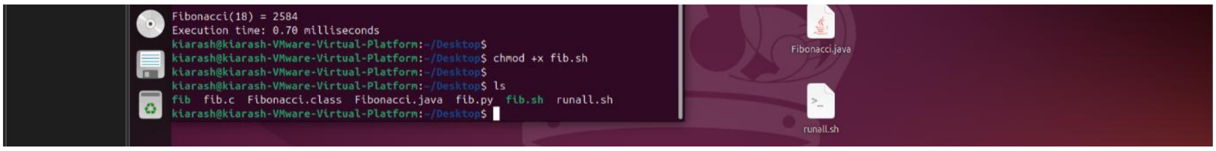
```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ gcc -o fib fib.c

kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
fib  fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ 
```

For fib.py:



```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
fib  fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ python3 fib.py

Fibonacci(18) = 2584
Execution time: 0.70 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
```

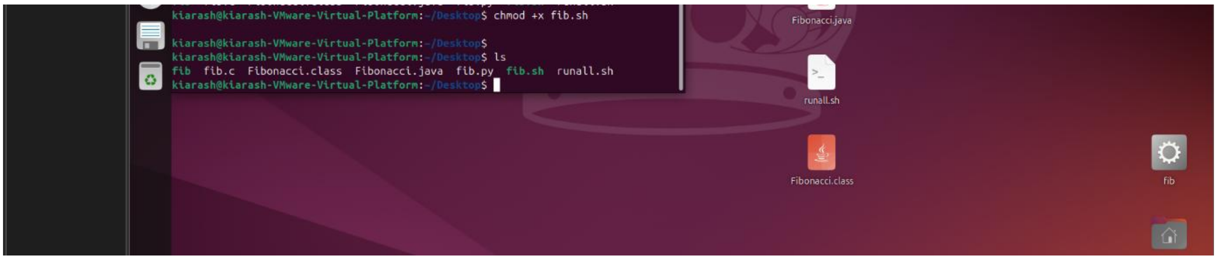
For fib.sh:



```
Fibonacci(18) = 2584
Execution time: 0.70 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ chmod +x fib.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
fib  fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ 
```

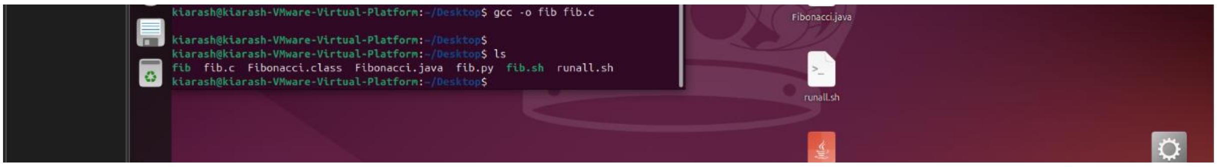
- **Make them executable:**

For fib.sh:



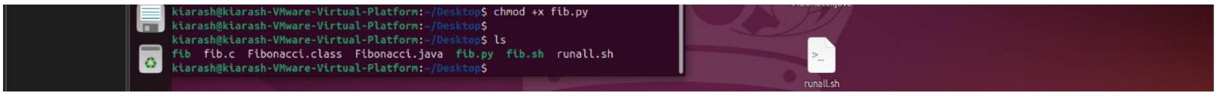
```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ chmod +x fib.sh

kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
fib  fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ 
```

For fib.c:



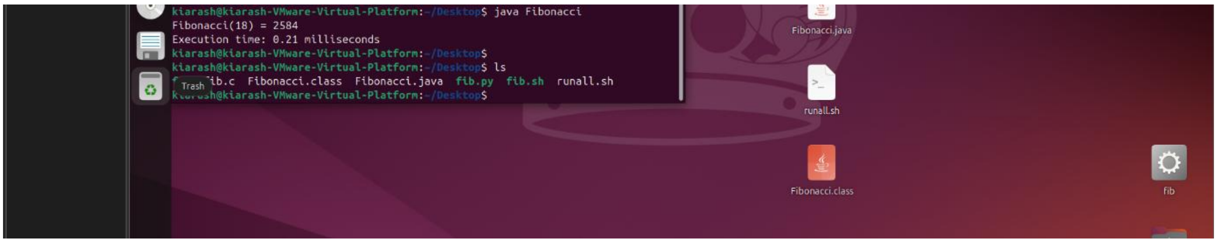
```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ gcc -o fib fib.c

kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
fib  fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
```

For fib.py:



```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ chmod +x fib.py
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
fib  fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
```

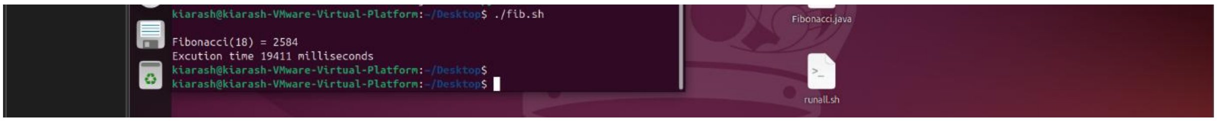
For Fibonacci.class (Complied file):



```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ java Fibonacci
Fibonacci(18) = 2584
Execution time: 0.21 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ls
f      ib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  runall.sh
k    sh@kiarash-VMware-Virtual-Platform:~/Desktop$
```

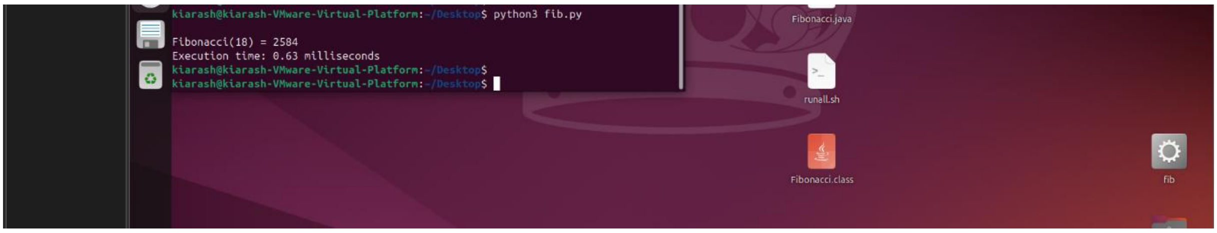
- **Run them:**

For fib.sh:



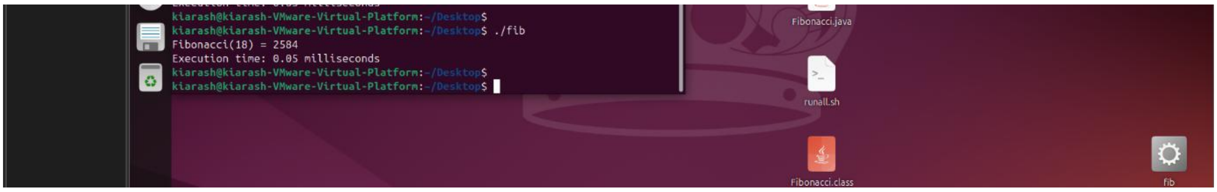
```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ./fib.sh

Fibonacci(18) = 2584
Excution time 19411 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ █
```

For fib.py:



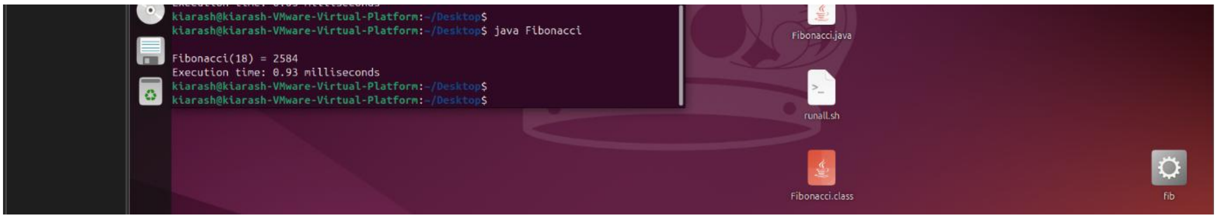
```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ python3 fib.py

Fibonacci(18) = 2584
Execution time: 0.63 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ █
```

For fib.C:



```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ./fib
Fibonacci(18) = 2584
Execution time: 0.05 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ █
```

For Fibonacci.class:



```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ java Fibonacci

Fibonacci(18) = 2584
Execution time: 0.93 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
```

- **Which (compiled) source code file performs the calculation the fastest?:**

+ The C program (fib) is the fastest. It performs the calculation **in 0.05 milliseconds.**

+ Java Fibonacci.class **is 0.93 milliseconds.**

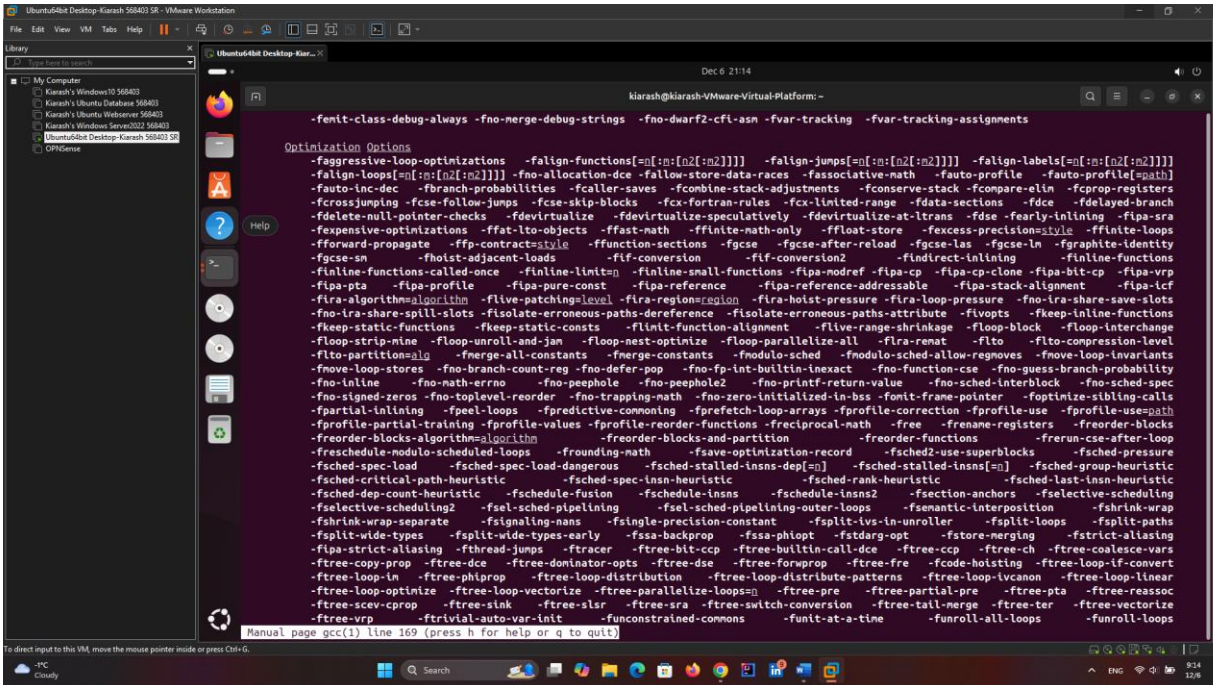
+ Bash Script (fib.sh) **is 19.6 seconds.**

Assignment 4.4: Optimize

Take relevant screenshots of the following commands:

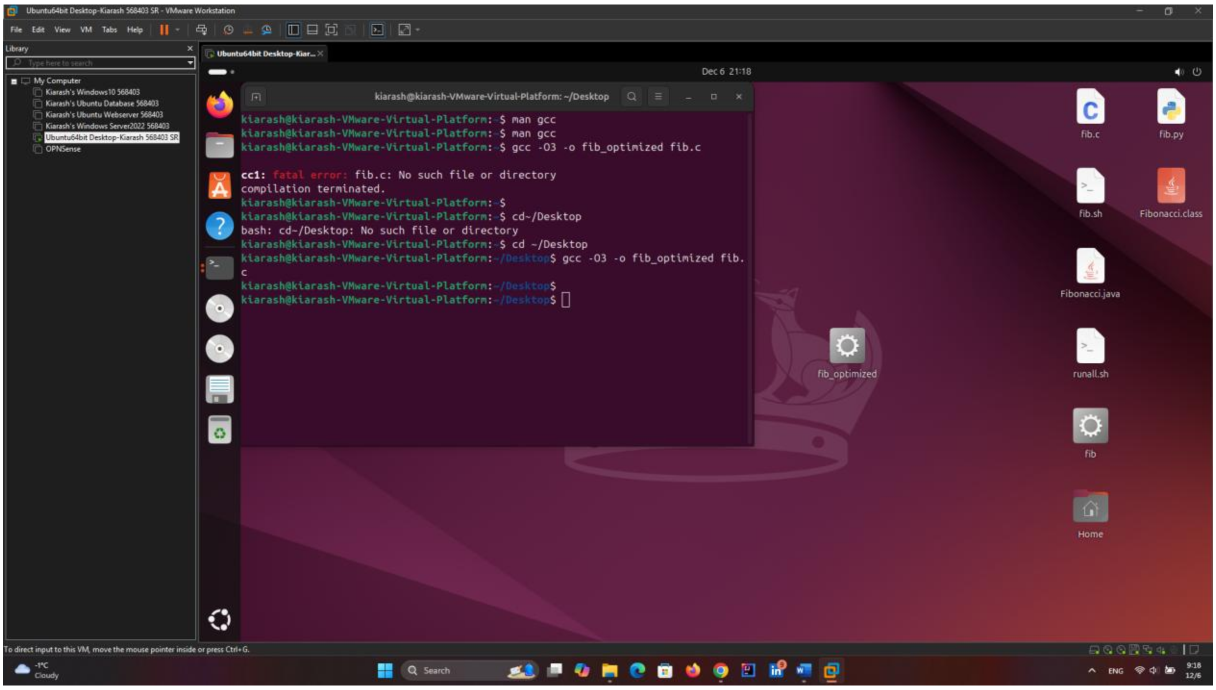
a) Figure out which parameters you need to pass to **the gcc** compiler so that the compiler performs a number of optimizations that will ensure that the compiled source code will run faster. **Tip!** The parameters are usually a letter followed by a number. Also read **page 191** of your book, but find a better optimization in the man pages. Please note that Linux is case sensitive.

Optimization Options:



Ubuntu64bit Desktop-Kiarash 568403 SR - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Kiarash's Windows10 568403
Kiarash's Ubuntu Database 568403
Kiarash's Ubuntu Webserver 568403
Kiarash's Windows Server2022 568403
Ubuntu64bit Desktop-Kiarash 568403 SR
OPNSense
Ubuntu64bit Desktop-Kiar...
Dec 6 21:14
kiarash@kiarash-VMware-Virtual-Platform: ~
Help
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Search
ENG
9:14
12/6

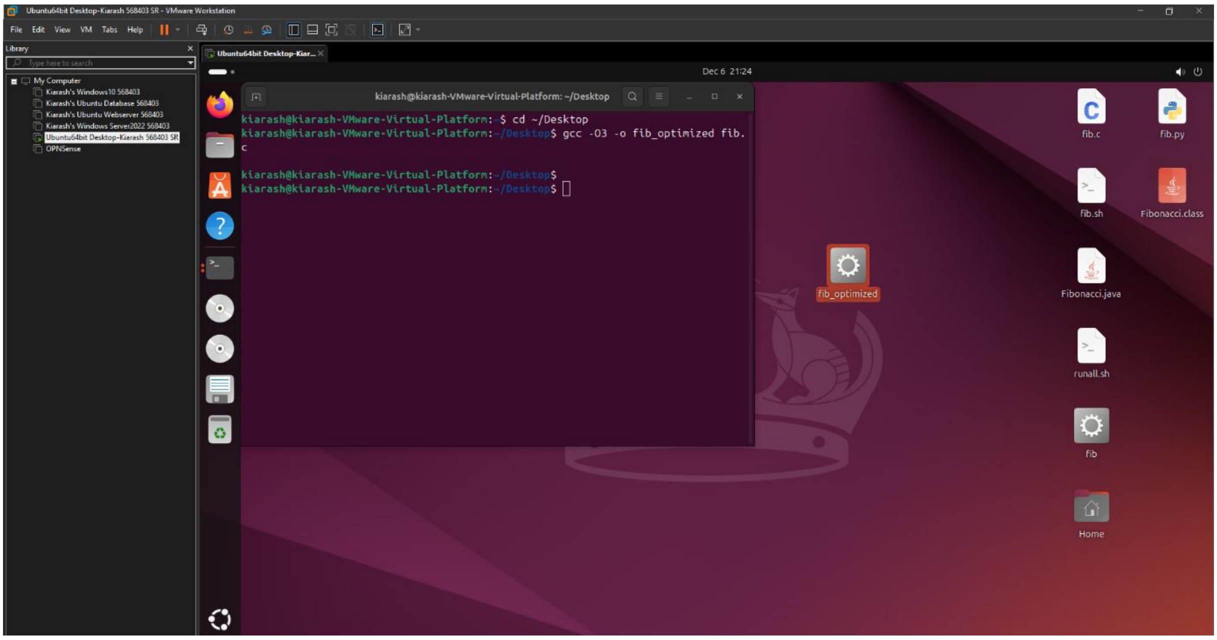
```
           -femit-class-debug-always -fno-merge-debug-strings  -fno-dwarf2-cfi-asm -fvar-tracking  -fvar-tracking-assignments

       Optimization Options
           -faggressive-loop-optimizations   -falign-functions[=n[:m:[n2[:m2]]]]   -falign-jumps[=n[:m:[n2[:m2]]]]  -falign-labels[=n[:m:[n2[:m2]]]]
           -falign-loops[=n[:m:[n2[:m2]]]] -fno-allocation-dce -fallow-store-data-races  -fassociative-math   -fauto-profile   -fauto-profile[=path]
           -fauto-inc-dec   -fbranch-probabilities  -fcaller-saves  -fcombine-stack-adjustments   -fconserve-stack -fcompare-elim  -fcprop-registers
           -fcrossjumping -fcse-follow-jumps  -fcse-skip-blocks   -fcx-fortran-rules  -fcx-limited-range  -fdata-sections   -fdce   -fdelayed-branch
           -fdelete-null-pointer-checks   -fdevirtualize   -fdevirtualize-speculatively  -fdevirtualize-at-ltrans  -fdse -fearly-inlining  -fipa-sra
           -fexpensive-optimizations  -ffat-lto-objects  -ffast-math   -ffinite-math-only   -ffloat-store   -fexcess-precision=style  -ffinite-loops
           -fforward-propagate   -ffp-contract=style   -ffunction-sections  -fgcse   -fgcse-after-reload  -fgcse-las  -fgcse-lm  -fgraphite-identity
           -fgcse-sm        -fhoist-adjacent-loads        -fif-conversion        -fif-conversion2        -findirect-inlining         -finline-functions
           -finline-functions-called-once   -finline-limit=n  -finline-small-functions -fipa-modref -fipa-cp  -fipa-cp-clone -fipa-bit-cp  -fipa-vrp
           -fipa-pta    -fipa-profile     -fipa-pure-const    -fipa-reference      -fipa-reference-addressable    -fipa-stack-alignment     -fipa-icf
           -fira-algorithm=algorithm  -flive-patching=level -fira-region=region  -fira-hoist-pressure -fira-loop-pressure  -fno-ira-share-save-slots
           -fno-ira-share-spill-slots -fisolate-erroneous-paths-dereference  -fisolate-erroneous-paths-attribute  -fivopts   -fkeep-inline-functions
           -fkeep-static-functions   -fkeep-static-consts    -flimit-function-alignment    -flive-range-shrinkage  -floop-block   -floop-interchange
           -floop-strip-mine  -floop-unroll-and-jam   -floop-nest-optimize  -floop-parallelize-all   -flra-remat    -flto    -flto-compression-level
           -flto-partition=alg    -fmerge-all-constants   -fmerge-constants   -fmodulo-sched   -fmodulo-sched-allow-regmoves  -fmove-loop-invariants
           -fmove-loop-stores  -fno-branch-count-reg -fno-defer-pop   -fno-fp-int-builtin-inexact   -fno-function-cse  -fno-guess-branch-probability
           -fno-inline     -fno-math-errno     -fno-peephole   -fno-peephole2    -fno-printf-return-value    -fno-sched-interblock   -fno-sched-spec
           -fno-signed-zeros -fno-toplevel-reorder  -fno-trapping-math  -fno-zero-initialized-in-bss -fomit-frame-pointer   -foptimize-sibling-calls
           -fpartial-inlining   -fpeel-loops   -fpredictive-commoning  -fprefetch-loop-arrays -fprofile-correction -fprofile-use  -fprofile-use=path
           -fprofile-partial-training -fprofile-values -fprofile-reorder-functions -freciprocal-math   -free   -frename-registers   -freorder-blocks
           -freorder-blocks-algorithm=algorithm          -freorder-blocks-and-partition           -freorder-functions           -frerun-cse-after-loop
           -freschedule-modulo-scheduled-loops    -frounding-math       -fsave-optimization-record      -fsched2-use-superblocks       -fsched-pressure
           -fsched-spec-load      -fsched-spec-load-dangerous     -fsched-stalled-insns-dep[=n]    -fsched-stalled-insns[=n]   -fsched-group-heuristic
           -fsched-critical-path-heuristic          -fsched-spec-insn-heuristic           -fsched-rank-heuristic             -fsched-last-insn-heuristic
           -fsched-dep-count-heuristic    -fschedule-fusion     -fschedule-insns      -fschedule-insns2      -fsection-anchors    -fselective-scheduling
           -fselective-scheduling2    -fsel-sched-pipelining       -fsel-sched-pipelining-outer-loops        -fsemantic-interposition        -fshrink-wrap
           -fshrink-wrap-separate      -fsignaling-nans     -fsingle-precision-constant      -fsplit-ivs-in-unroller        -fsplit-loops      -fsplit-paths
           -fsplit-wide-types     -fsplit-wide-types-early      -fssa-backprop      -fssa-phiopt     -fstdarg-opt         -fstore-merging        -fstrict-aliasing
           -fipa-strict-aliasing   -fthread-jumps    -ftracer     -ftree-bit-ccp   -ftree-builtin-call-dce    -ftree-ccp     -ftree-ch   -ftree-coalesce-vars
           -ftree-copy-prop   -ftree-dce     -ftree-dominator-opts   -ftree-dse     -ftree-forwprop     -ftree-fre     -fcode-hoisting   -ftree-loop-if-convert
           -ftree-loop-im    -ftree-phiprop      -ftree-loop-distribution      -ftree-loop-distribute-patterns     -ftree-loop-ivcanon    -ftree-loop-linear
           -ftree-loop-optimize  -ftree-loop-vectorize  -ftree-parallelize-loops=n   -ftree-pre     -ftree-partial-pre      -ftree-pta     -ftree-reassoc
           -ftree-scev-cprop     -ftree-sink      -ftree-slsr     -ftree-sra   -ftree-switch-conversion    -ftree-tail-merge   -ftree-ter     -ftree-vectorize
           -ftree-vrp        -ftrivial-auto-var-init       -funconstrained-commons        -funit-at-a-time          -funroll-all-loops            -funroll-loops
 Manual page gcc(1) line 169 (press h for help or q to quit)
```



Ubuntu64bit Desktop-Kiarash 568403 SR - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Kiarash's Windows10 568403
Kiarash's Ubuntu Database 568403
Kiarash's Ubuntu Webserver 568403
Kiarash's Windows Server2022 568403
Ubuntu64bit Desktop-Kiarash 568403 SR
OPNSense
Ubuntu64bit Desktop-Kiar...
Dec 6 21:18
kiarash@kiarash-VMware-Virtual-Platform: ~/Desktop
fib.c
fib.py
fib.sh
Fibonacci.class
Fibonacci.java
fib_optimized
runall.sh
fib
Home
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Search
ENG
9:18
12/6

```
kiarash@kiarash-VMware-Virtual-Platform:~$ man gcc
kiarash@kiarash-VMware-Virtual-Platform:~$ man gcc
kiarash@kiarash-VMware-Virtual-Platform:~$ gcc -O3 -o fib_optimized fib.c

cc1: fatal error: fib.c: No such file or directory
compilation terminated.
kiarash@kiarash-VMware-Virtual-Platform:~$
kiarash@kiarash-VMware-Virtual-Platform:~$ cd~/Desktop
bash: cd~/Desktop: No such file or directory
kiarash@kiarash-VMware-Virtual-Platform:~$ cd ~/Desktop
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ gcc -O3 -o fib_optimized fib.
c
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ 
```

b) Compile **fib.c** again with the optimization parameters:

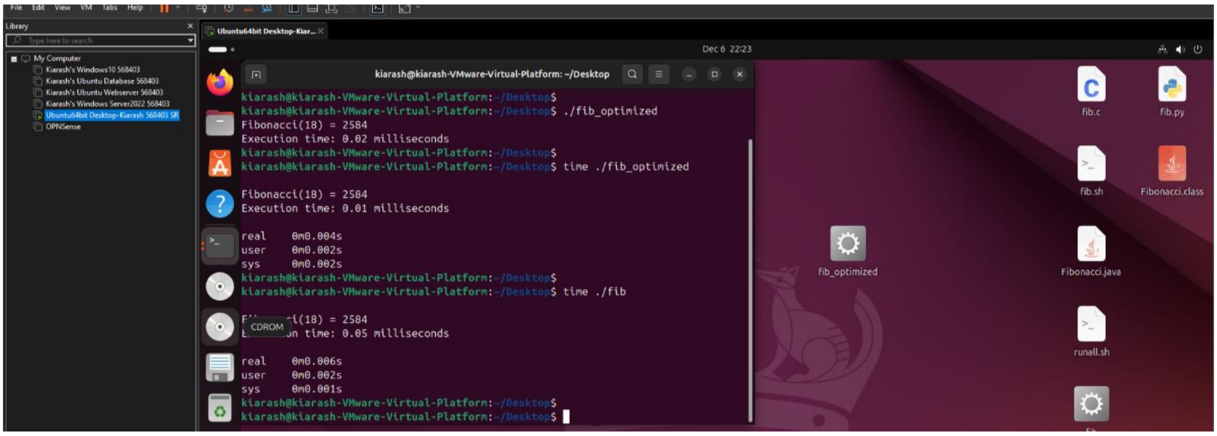


```
kiarash@kiarash-VMware-Virtual-Platform:~$ cd ~/Desktop
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ gcc -O3 -o fib_optimized fib.
c

kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ 
```

Ubuntu64bit Desktop-Kiarash 568403 SR - VMware Workstation
File
Edit
View
VM
Tabs
Help
Library
Type here to search
My Computer
Kiarash's Windows10 568403
Kiarash's Ubuntu Database 568403
Kiarash's Ubuntu Webserver 568403
Kiarash's Windows Server2022 568403
Ubuntu64bit Desktop-Kiarash 568403 SR
OPNSense
Ubuntu64bit Desktop-Kiar...
Dec 6 21:24
kiarash@kiarash-VMware-Virtual-Platform: ~/Desktop
fib.c
fib.py
fib.sh
Fibonacci.class
fib_optimized
Fibonacci.java
runall.sh
fib
Home

c) Run the newly compiled program. Is it true that it now performs the calculation faster?

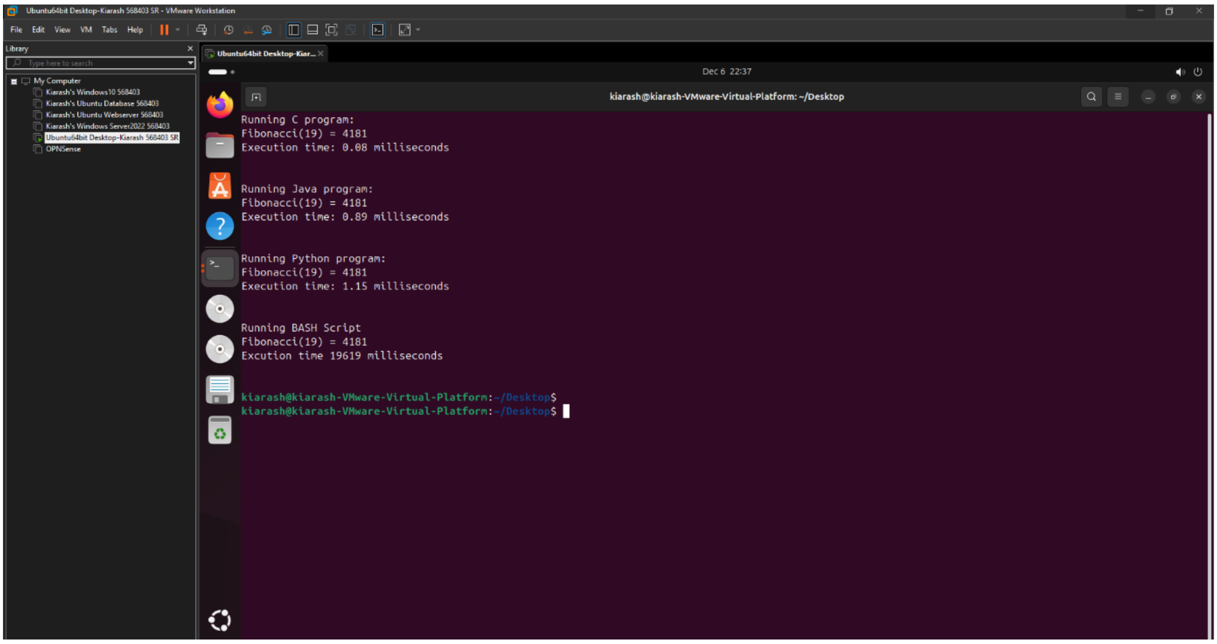
According to the picture, " ./fib optimized " is faster than non-optimized version:



```
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ ./fib_optimized
Fibonacci(18) = 2584
Execution time: 0.02 milliseconds
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ time ./fib_optimized

Fibonacci(18) = 2584
Execution time: 0.01 milliseconds

real    0m0.004s
user    0m0.002s
sys     0m0.002s
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ time ./fib

F      i(18) = 2584
E      on time: 0.05 milliseconds

real    0m0.006s
user    0m0.002s
sys     0m0.001s
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$ 
```

File
Edit
View
VM
Tabs
Help
Library
Type here to search
My Computer
Kiarash's Windows10 568403
Kiarash's Ubuntu Database 568403
Kiarash's Ubuntu Webserver 568403
Kiarash's Windows Server2022 568403
Ubuntu64bit Desktop-Kiarash 568403 SR
OPNSense
Ubuntu64bit Desktop-Kiar...
Dec 6 22:23
kiarash@kiarash-VMware-Virtual-Platform: ~/Desktop
CDROM
fib.c
fib.py
fib.sh
Fibonacci.class
fib_optimized
Fibonacci.java
runall.sh

d) Edit the file **runall.sh**, so you can perform all four calculations in a row using this Bash script. So the (compiled/interpreted) C, Java, Python and Bash versions of Fibonacci one after the other.



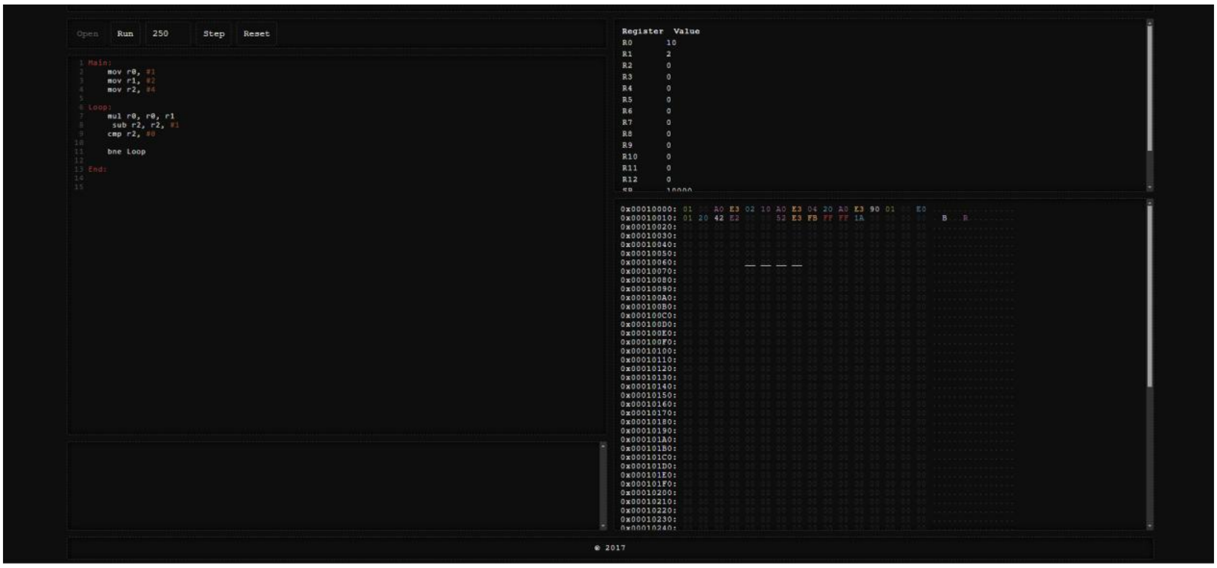
Ubuntu64bit Desktop-Kiarash 568403 SR - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Kiarash's Windows10 568403
Kiarash's Ubuntu Database 568403
Kiarash's Ubuntu Webserver 568403
Kiarash's Windows Server2022 568403
Ubuntu64bit Desktop-Kiarash 568403 SR
OPNSense
Ubuntu64bit Desktop-Kiar...
Dec 6 22:37
kiarash@kiarash-VMware-Virtual-Platform: ~/Desktop

```
Running C program:
Fibonacci(19) = 4181
Execution time: 0.08 milliseconds


Running Java program:
Fibonacci(19) = 4181
Execution time: 0.89 milliseconds


Running Python program:
Fibonacci(19) = 4181
Execution time: 1.15 milliseconds


Running BASH Script
Fibonacci(19) = 4181
Excution time 19619 milliseconds


kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
kiarash@kiarash-VMware-Virtual-Platform:~/Desktop$
```

Bonus point assignment – week 4

Like the factorial example, you can also implement the calculation of a power of 2 in assembly. For example you want to calculate $2^4 = 16$. Use iteration to calculate the result. Store the result in r0.

This the screenshot: Kiarash Delavar – 568403 – SR



```
Main:
    mov r0, #1
    mov r1, #2
    mov r2, #4

Loop:
    mul r0, r0, r1
     sub r2, r2, #1
    cmp r2, #0

    bne Loop

End:
```

```
Register  Value
R0        10
R1        2
R2        0
R3        0
R4        0
R5        0
R6        0
R7        0
R8        0
R9        0
R10       0
R11       0
R12       0
SP        10000
```

```
0x00010000: 01 00 A0 E3 02 10 A0 E3 04 20 A0 E3 90 01 00 E0
0x00010010: 01 20 42 E2 00 00 52 E3 FB FF FF 1A 00 00 00 00
0x00010020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000100F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000101A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000101B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000101C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000101D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000101E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000101F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00010240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

This is the code that I wrote for that:

```
Main:
 mov r0, #1
 mov r1, #2
 mov r2, #4

Loop:
mul r0, r0, r1
sub r2, r2, #1
cmp r2, #0

bne Loop

End:
```

Complete the code. See the PowerPoint slides of week 4.

Screenshot of the completed code here.

Ready? Save this file and export it as a pdf file with the name: **week4.pdf**